

**IMPLEMENTASI DATA MINING UNTUK MEMPREDIKSI KELAYAKAN PERMINTAAN
PINJAMAN NASABAH DI LEMBAGA KEUANGAN
(Studi Kasus di Koperasi Simpan Pinjam JASA Kota Pekalongan)**

Mukhammad Yunan Helmy, Drs. Kushartantya M.Ikomp, Nurdin Bahtiar S.Si., M.T.

Ilmu Komputer / Informatika FSM Universitas Diponegoro

m.yunan.helmy@gmail.com, kushartantya@ilkom.undip.ac.id, nurdinbahtiar@gmail.com

Abstrak

Perkembangan teknologi informasi saat ini memungkinkan penyimpanan data dalam skala yang sangat besar. Perkembangan ini telah merambah ke berbagai bidang, termasuk perbankan, sehingga pemanfaatan teknologi informasi di lembaga keuangan sudah tidak asing lagi di kalangan masyarakat luas. *Data mining* merupakan kegiatan yang meliputi pengumpulan dan pemakaian data historis untuk menemukan keteraturan, pola, atau hubungan dalam data berukuran besar. Dengan memanfaatkan data nasabah dan data pinjaman, *data mining* dapat membantu keputusan strategis dalam memprediksi kelayakan pinjaman untuk nasabah di sebuah lembaga keuangan. Algoritma yang digunakan untuk membangun *decision tree* yaitu algoritma ID3 (*Iterative Dichotomizer 3*). Algoritma ID3 digunakan untuk mengkategorikan data menjadi beberapa kelas. Dari hasil percobaan dengan 20 data baru dihasilkan akurasi sebesar 85%. Hasil yang diperoleh dapat digunakan sebagai bahan masukan bagi instansi terkait dalam uji kelayakan pinjaman dari nasabahnya.

Kata kunci : *data mining*, prediksi, kelayakan pinjaman, *decision tree*, algoritma ID3.

Abstract

Developing of information technology give some options to save data in large scale recently. This development has already been expanding in many fields, such as banking, then there is no strangeness anymore among the public. Data mining is an activity that collect and use historical data to find harmony, pattern, and relationship in a large data. By using customer data dan loan data, data mining could support to decide the appropriate strategic in order to make a loan feasibility prediction in financial institution. Algorithm which is used to create a decision tree is ID3 Algorithm (*Iterative Dichotomizer 3*). ID3 algorithm is used to category data into many classes. From the experiment with 20 new data shown the accuracy is 85%. The result of this research can be used as a recommendation for institution in order to check of loan feasibility from customers.

Keyword : data mining, prediction, loan feasibility, decision tree, ID3 algorithm.

1. Pendahuluan

Kemajuan teknologi saat ini memungkinkan penyimpanan data dalam skala yang sangat besar. Data yang sangat banyak tersebut diubah menjadi informasi dan informasi diubah menjadi pengetahuan. Proses menganalisis data, mengubah data menjadi informasi, dan menemukan pola yang dapat dijadikan pengetahuan merupakan pengertian dari *data mining* [6].

Sebagai cabang ilmu di bidang komputer, cukup banyak penerapan yang dilakukan oleh *data mining*. *Data mining* berkembang dan diterapkan dalam dunia perbankan karena ditunjang oleh ketersediaan data.

Koperasi Simpan Pinjam JASA merupakan koperasi yang bergerak dibidang simpan pinjam.

Tidak semua anggota Koperasi Simpan Pinjam JASA menyetorkan tanggungan pinjamannya tepat pada jatuh tempo. Jika diamati, maka terdapat pola nasabah mana yang menunggak dan nasabah mana yang tidak menunggak. Pola tersebut dapat dilihat melalui data yang telah ada di Koperasi Simpan Pinjam JASA dengan menggunakan teknik *data mining*.

Data mining berarti menggali pengetahuan dari data yang berjumlah banyak [3]. *Data mining* dapat diterapkan untuk menggali informasi dari tumpukan data tersebut. Salah satu penerapannya yaitu mencari pola pinjaman nasabah melalui data pinjaman nasabah yang telah tersedia. Data pinjaman nasabah ditransformasikan, dibersihkan dan dicari polanya sehingga dapat diketahui

bagaimana pola nasabah yang menunggak dalam peminjaman.

Metode pohon keputusan (*decision tree*) merupakan metode yang mengubah fakta yang sangat besar menjadi pohon keputusan yang merepresentasikan aturan. Aturan dapat dengan mudah dipahami dengan bahasa alami dan dapat diekspresikan dalam bentuk basis data seperti *Structured Query Language* untuk mencari *record* pada kategori tertentu [7].

Di dalam data operasional dan *data warehouse* Koperasi Simpan Pinjam JASA terdapat data anggota yang melakukan pinjaman. Data tersebut dapat dianalisis dan dicari aturannya menggunakan metode pohon keputusan (*decision tree*). Oleh karena itu, dengan memanfaatkan teknik *data mining* tersebut dapat diprediksi nasabah mana yang layak diterima pinjamannya dan tidak layak diterima pinjamannya.

2. Landasan Teori

2.1. Pengertian Data Mining

Data mining adalah proses menganalisis data dan menemukan pola tersembunyi secara otomatis atau semiotomatis [12]. Menurut Gartner Group, *data mining* adalah proses mencari hubungan, pola, dan *trend* melalui data yang besar, menggunakan teknologi pengenalan pola, teknik statistika, dan matematika [8]. *Data mining* adalah kegiatan yang meliputi pengumpulan, pemakaian data historis untuk menemukan keteraturan, pola atau hubungan dalam set data berukuran besar [15]. Pola atau hubungan digunakan sebagai alat untuk menjelaskan data dan memprediksi data yang baru [4].

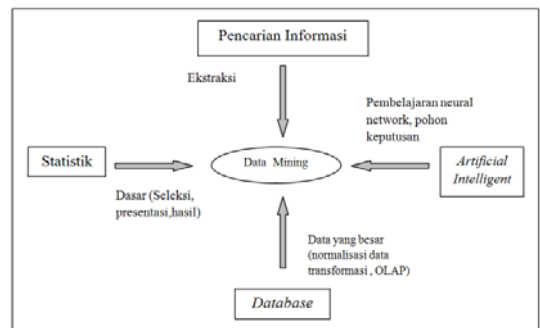
Ada beberapa definisi lain dari *data mining*. *Data mining* adalah proses menganalisis data yang sangat besar untuk menemukan hubungan dan merepresentasikan data yang berguna dan dapat dipahami untuk pemiliknya [8].

Secara sederhana, *data mining* berarti menggali pengetahuan dari data yang berjumlah banyak [3]. *Data mining* adalah proses yang menggunakan teknik statistik, matematika, kecerdasan buatan, dan *machine learning* untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai *database* besar [7].

Data mining sering disebut *knowledge discovery in database* (KDD), adalah kegiatan yang meliputi pengumpulan, pemakaian data historis untuk menemukan keteraturan, pola atau hubungan dalam set data yang berukuran besar. Keluaran dari *data mining* ini bisa dipakai untuk memperbaiki pengambilan keputusan di masa depan sehingga

istilah *pattern recognition* jarang digunakan karena termasuk dalam bagian *data mining* [15].

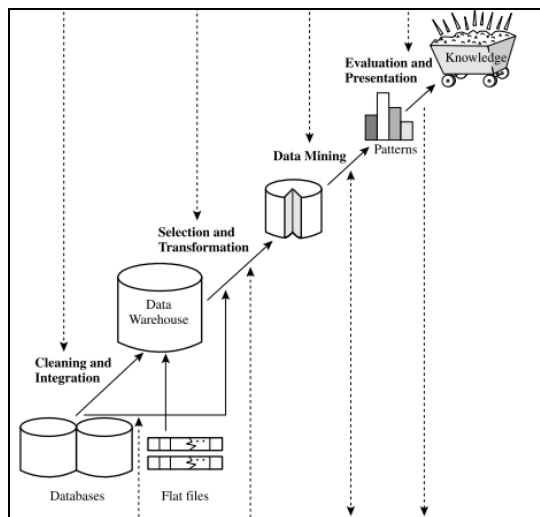
Data mining bukanlah suatu bidang ilmu yang baru. Salah satu kesulitan mendefinisikan *data mining* adalah kenyataan bahwa *data mining* mewarisi banyak aspek dan teknik dari bidang-bidang ilmu yang sudah mapan terlebih dahulu. Gambar 1 merupakan bidang ilmu yang menjadi akar panjang dari *data mining*. Beberapa bidang ilmu tersebut seperti kecerdasan buatan, *machine learning*, statistik, *database*, dan juga *information retrieval* [7].



Gambar 1 Bidang Ilmu *Data Mining*

2.2. Tahapan Data Mining

Istilah *data mining* dan *Knowledge Discovery in Database* (KDD) sering kali digunakan secara bergantian untuk menjelaskan proses penggalian informasi tersembunyi dalam suatu basis data yang besar. Kedua istilah tersebut memiliki konsep yang berbeda, tetapi berkaitan satu sama lain. Salah satu tahapan dari KDD adalah *data mining*. Tahapan *data mining* ditunjukkan pada Gambar 2.



Gambar 2 Tahapan *Data Mining*

Tahapan *data mining* ada 7 yaitu : [3]

1. *Data Cleaning*
Data cleaning merupakan proses menghilangkan *noise*, data yang tidak konsisten, dan data yang tidak relevan.
2. *Data Integration*
Data integration merupakan proses menggabungkan data dari berbagai data sumber (*data source*) ke dalam *database* yang akan digunakan untuk proses penggalian data.
3. *Data Selection*
Data selection merupakan proses pemilihan data yang digunakan untuk proses penggalian data.
4. *Data Transformation*
Data Transformation merupakan proses mentransformasikan dan mengkonsolidasikan data untuk digunakan dalam proses *mining*.
5. *Data Mining*
Data mining merupakan proses utama mencari pengetahuan atau pola dari informasi tersembunyi dari *database*.
6. *Pattern Evaluation*
Pattern Evaluation merupakan proses mengidentifikasi pola yang telah didapat.
7. *Knowledge Presentation*
Knowledge Presentation merupakan visualisasi dan presentasi pengetahuan atau pola yang telah didapat kepada pengguna.

2.3. Pengertian *Decision Tree*

Decision Tree atau pohon keputusan merupakan teknik mengubah fakta yang besar menjadi pohon keputusan yang merepresentasikan aturan. Pohon keputusan terdiri dari simpul (*nodes*) dan cabang (*branches*) yang menyambungkan simpul. Simpul terbawah dari pohon keputusan disebut daun (*leaves*) dan mengindikasikan sebagai kelas. Sedangkan simpul paling atas dari pohon keputusan disebut akar (*root*). Semua simpul kecuali daun disebut simpul keputusan (*decision nodes*) karena digunakan untuk menggolongkan keputusan [6].

Decision tree adalah pendekatan *divide and conquer* dalam proses pembelajaran dari kumpulan data dan direpresentasikan secara alami [4]. Prinsip dari *decision tree* yaitu membagi data secara rekursif ke dalam sub himpunan dimana setiap sub himpunan tersebut memiliki kondisi homogen dari variabel target (atribut yang diprediksi) yang disebut sebagai kelas [12]. Ada beberapa keuntungan menggunakan *decision tree* dibanding dengan lainnya yaitu *decision tree* mudah dibangun dan mudah diinterpretasikan. Prediksi dari *decision tree* juga efisien [12].

Semua algoritma pembentuk pohon keputusan berdasarkan konsep belajar algoritma Hunt. Algoritma pembentuk pohon keputusan menggunakan metode manusia ketika belajar sesuatu yang sederhana, yaitu dengan metode *divide and conquer*. Data dibagi menjadi himpunan *S* yang berisi sejumlah *n data training* yang termasuk dari kelas *c* kemudian dibagi menjadi sub-himpunan yang menghasilkan partisi yang hanya berisi *data training* dari satu kelas saja.

Secara garis besar, algoritma tersebut mempunyai langkah sebagai berikut [6] :

Diberikan : Himpunan data *training S*

1. Pilih pembeda yang paling signifikan
2. Bagi seluruh himpunan *S* mulai dari *root* ke dalam beberapa sub-himpunan menggunakan cara yang dipilih.
3. Lakukan langkah 1 dan 2 secara rekursif untuk menemukan pembeda yang paling signifikan dari sub-himpunan dan bagi ke dalam sub-himpunan lagi. Jika sub-himpunan sudah berisi satu kelas saja maka hentikan perulangan. Jika sebaliknya, lakukan langkah 3 secara rekursif.

Hasil : Pohon keputusan yang bisa diambil aturannya

Berdasarkan algoritma Hunt, Quinlan membangun algoritma yang disebut ID3 (*Iterative Dichotomizer 3*) yang menggunakan Entropi Shannon sebagai kriteria untuk memilih pembeda yang paling signifikan, rumus entropi sebagai berikut:

$$\text{Entropy}(S) = - \sum_{i=1}^n p_i \cdot \log_2(p_i)$$

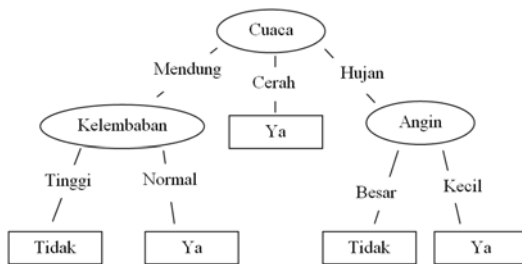
dimana p_i adalah proporsi dari data yang termasuk dalam kelas ke-*i*.

Pada penyusunan pohon keputusan, pemilihan atribut sebagai simpul dapat dilakukan dengan melihat atribut tersebut dapat memisahkan obyek menurut kelasnya. Secara heuristik atribut yang dipilih adalah atribut yang paling bersih (*purest*). Jika dalam satu cabang anggotanya berasal dari satu kelas maka cabang tersebut disebut *pure*. Semakin *pure* suatu cabang semakin baik. Ukuran *purity* dinyatakan dengan tingkat *impurity*. Salah satu kriteria *impurity* adalah *information gain*. *Information gain* yang paling besar menandakan atribut tersebut dapat memecah obyek menjadi beberapa kelas,

$$\text{Information Gain}(S, F_j) = \text{Entropy}(S) - \sum_{i=1}^n \frac{|S_{V_i}|}{|S|} \cdot \text{Entropy}(S_{V_i})$$

dimana V_{F_j} adalah himpunan dari semua kemungkinan nilai dari F_j dan S_{V_i} adalah sub-himpunan dari *S*, dimana setiap F_j mempunyai V_i .

Contoh model pohon keputusan untuk mengambil keputusan bermain tenis ditunjukkan pada Gambar 3. Ada tiga atribut untuk menentukan keputusan bermain tenis atau tidak. Atribut tersebut yaitu cuaca, kelembaban, dan angin. Pemilihan simpul akar dengan menggunakan *information gain* terbesar yang berarti simpul yang *purest*. Pemilihan simpul selanjutnya dilakukan secara rekursif hingga menemukan simpul yang hanya berisi obyek dalam satu kelas. Simpul yang paling bawah (daun) merupakan kelas dari obyek yang akan digunakan sebagai prediksi keputusan.



Gambar 3 Contoh Pohon Keputusan

3. Analisis dan Perancangan

3.1. Analisis Data Mining

Adanya kemungkinan penunggakan atau kemacetan pelunasan pinjaman menjadi permasalahan dalam penyeleksian nasabah baru. Proses penyeleksian pinjaman nasabah baru masih memakan waktu yang lama. Jadi, diperlukan sebuah perangkat lunak untuk memberikan rekomendasi awal terkait kelayakan pinjaman.

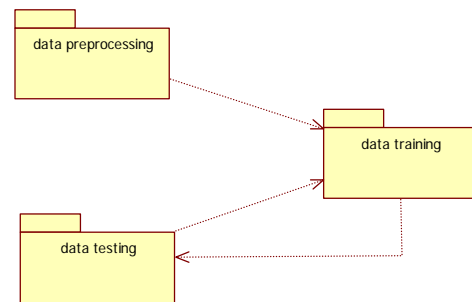
Data nasabah lama yang sudah ada dapat dianalisis dan hasil analisisnya dapat berguna untuk prediksi atau rekomendasi penyeleksian. Oleh karena itu dilakukan *data mining* guna memprediksi kelayakan peminjaman dari nasabah.

Kelayakan permintaan pinjaman dapat dilihat dari data pribadi nasabah dan data permintaan pinjaman. Data tersebut dicari pola atau aturannya sehingga dapat digunakan untuk memprediksi nasabah mana yang menunggak dan tidak menunggak dengan teknik *decision tree*.

Data yang digunakan yaitu data yang sudah disediakan oleh pihak Koperasi Simpan Pinjam JASA. Data tersebut akan melalui tahap *data preprocessing* terlebih dahulu untuk dibersihkan, ditransformasikan dan dipilih yang kemudian akan digunakan sebagai *data set*. *Data set* tersebut menjadi data yang digunakan untuk pelatihan (*data training*). Dari *data training* tersebut didapatkan sebuah *decision tree*. *Decision tree* tersebut digunakan untuk memprediksi atau menganalisis

nasabah baru apakah nasabah tersebut layak diterima atau tidak permintaan pinjamannya.

Tahapan tersebut merupakan proses bisnis yang ada dalam pengimplementasian *data mining* di Koperasi Simpan Pinjam JASA. Tahapan tersebut mempunyai hubungan yang ditunjukkan pada gambar 4.



Gambar 4 Hubungan Tiga Proses Bisnis

3.2. Sumber Data dan Data yang Digunakan

Data yang digunakan sebagai sumber data yaitu data dari Koperasi Simpan Pinjam JASA. Data tersebut sudah ditransformasikan oleh pihak Koperasi Simpan Pinjam JASA untuk kepentingan penelitian ini. Data tersebut diambil dari data operasional Koperasi Simpan Pinjam JASA dan *data warehouse* Koperasi Simpan Pinjam JASA. Data tersebut berjumlah 20944 *record*.

Dalam penelitian ini, tidak semua atribut digunakan. *Data set* akan dibersihkan terlebih dahulu dan atribut – atribut yang kurang relevan tidak akan digunakan dalam analisis. Jumlah *data set* yang digunakan untuk *data training* yaitu 3082 baris *record*. Atribut tersebut dipilih berdasarkan rekomendasi dari pihak Kospin JASA dan relevansi atribut dengan kasus. Tabel 1 merupakan atribut – atribut yang dipilih beserta keterangannya.

Tabel 1 Atribut *data set* yang digunakan

No	Nama Atribut	Keterangan
1	Stat_ang	Status keanggotaan
2	Jenis_kel	Jenis kelamin nasabah
3	Agama	Agama nasabah
4	Pendidikan	Pendidikan nasabah
5	Akte	Nomor akte
6	Pekerjaan	Pekerjaan nasabah
7	Pdpt_bln	Pendapatan per bulan dari nasabah
8	Plafond	Jumlah permintaan pinjaman
9	Jw_bulan	Jangka waktu peminjaman dalam hitungan bulan
10	Keputusan	Status nasabah layak

No	Nama Atribut	Keterangan
		diterima tidak , berdasarkan status pernah menunggak atau tidak

3.3. Deskripsi Perangkat Lunak

Perangkat lunak yang dibangun berbasis *desktop* dengan nama MINER's. MINER's dibangun dengan menggunakan bahasa pemrograman Java dan memanfaatkan DBMS *SQL Server 2005*. Pengguna MINER's yaitu pihak Koperasi Simpan Pinjam JASA khususnya bagian pinjaman. MINER's berfungsi untuk memprediksi kelayakan peminjaman dari nasabah Koperasi Simpan Pinjam JASA.

3.4. Spesifikasi Kebutuhan Perangkat Lunak

Spesifikasi kebutuhan fungsional perangkat lunak merujuk pada kebutuhan akan perancangan *data mining*. Berdasarkan dari proses bisnis yang telah dirumuskan, spesifikasi kebutuhan fungsional yang dapat dirumuskan yaitu :

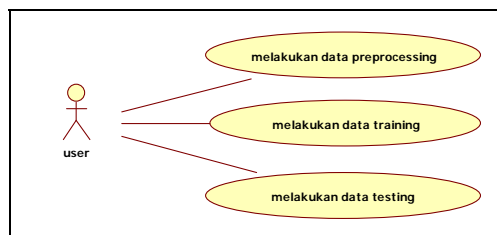
1. Dapat melakukan pra-proses data. Pra-proses data yaitu pembersihan data (*data cleaning*), pemilihan variabel (*data selection*), transformasi dan diskritisasi (*data transformation*).
2. Dapat membuat *decision tree* dengan algoritma ID3.
3. Dapat memberikan rekomendasi atau prediksi terhadap data baru berdasarkan dari aturan yang terdapat di *decision tree*.

3.5. Diagram Use Case

Proses bisnis yang dihasilkan yaitu :

1. Melakukan *Data preprocessing*
2. Melakukan *Data training*
3. Melakukan *Data testing*

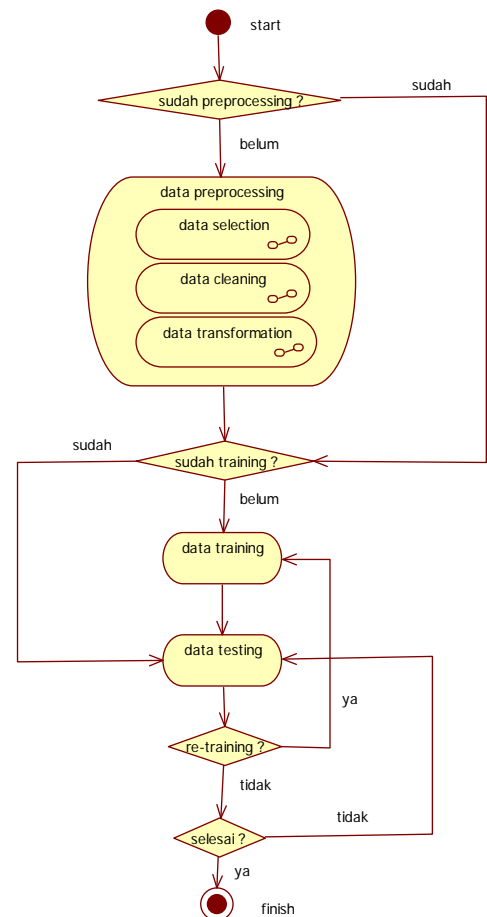
Diagram *use case* ditunjukkan pada gambar 5.



Gambar 5 Diagram Use Case

3.6. Diagram Aktifitas

Detail proses bisnis disajikan dengan menggunakan diagram aktifitas untuk menunjukkan bagaimana proses bisnis dilakukan. Proses bisnis dimulai dengan *data preprocessing*. Jika sudah menjalankan *data preprocessing* maka dapat menjalankan *data training*. Jika sudah menjalankan *data training* maka dapat menjalankan *data testing*. Setelah proses *data testing* pengguna dapat menjalankan *data training* lagi jika prediksi tidak diketahui. Gambar 6 merupakan diagram aktifitas perangkat lunak.



Gambar 6 Diagram Aktifitas

3.7. Data Preprocessing

Data preprocessing merupakan tahap dimana *data set* yang diterima akan diproses terlebih dahulu sebelum dilakukan analisis atau dilakukan *training*. Ada tiga sub-proses dalam *data preprocessing* yaitu :

- a. *Data selection*

- Pada tahap ini dipilih atribut – atribut yang relevan dengan proses *mining*. Hasil dari atribut dapat dilihat pada tabel 1.
- b. *Data cleaning*
 Pada tahap ini dilakukan proses pembersihan data dari *noise* dan *missing values*. Pembersihan dilakukan dengan metode menghapus (mengabaikan) *record* yang memiliki isian – isian *noise* atau *missing values*. Metode pengabaian tersebut dilakukan karena banyaknya jumlah data yang diperoleh.
- c. *Data transformation* / diskritisasi
 Pada tahap ini nilai / *value* dari atribut akan diubah ke bentuk diskrit untuk dapat diproses dengan algoritma ID3. Perubahan yang dilakukan ditunjukkan pada tabel 2.

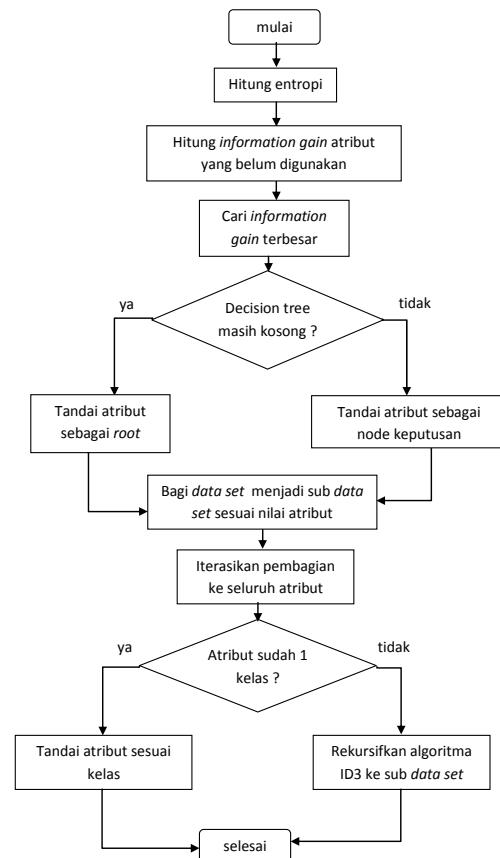
Tabel 2 Diskritisasi

Nama Atribut	transformasi (nilai awal → nilai diskrit)
Stat_ang	1 → ya 2 → bukan
Jenis_kel	1 → pria 2 → wanita
Agama	1 → islam 2 → protestan 3 → katholik 4 → hindu 5 → budha 6 → konghuchu 7 → lainnya
Pendidikan	1 → SD / SLTP 2 → SLTA 3 → D3 4 → S1 5 → S2
Akte	NULL → tidak ada NOT NULL → ada
Pekerjaan	Pegawai negeri → Pegawai negeri TNI / POLRI → TNI / POLRI Ibu rumah tangga → Ibu rumah tangga Profesional → Profesional Pegawai swasta → Pegawai swasta Wiraswasta → Wiraswasta Pelajar / mahasiswa → Pelajar / mahasiswa Lainnya → Lainnya
Pendapatan	0 – 2000000 → tipe a 2000001 – 4000000 → tipe b 4000001 – 6000000 → tipe c 6000001 – 8000000 → tipe d 8000001 – 10000000 → tipe e 10000001 – 15000000 → tipe f 15000001 - ... → tipe g
Plafond	0 – 25000000 → tipe 1 25000001 – 50000000 → tipe 2 50000001 – 75000000 → tipe 3 75000001 – 100000000 → tipe 4

Nama Atribut	transformasi (nilai awal → nilai diskrit)
	100000001 – 125000000 → tipe 5 125000001 – 150000000 → tipe 6 150000001 – 175000000 → tipe 7 175000001 – 200000000 → tipe 8 200000001 - ... → tipe 9
Waktu	0 – 12 bulan → sangat cepat 13 – 24 bulan → cepat 25 – 36 bulan → umum 37 – 48 bulan → lama 49 – 60 bulan → sangat lama
Keputusan	y → tidak layak n → layak

3.8. Data Training

Setelah *data set* telah melalui *preprocessing*, *data set* akan digunakan untuk *training*. Proses ini dilakukan dengan teknik *decision tree* menggunakan algoritma ID3. Kriteria pemilihan atribut dengan menggunakan entropi dan *information gain*. Gambar 7 menunjukkan pembentukan *decision tree* dengan algoritma ID3 yang dilakukan secara rekursif.



Gambar 7 Algoritma ID3

3.9. Data Testing

Pada tahap *data testing*, *decision tree* yang dihasilkan diuji dengan masukan data. Pengujian ini bertujuan untuk mengetahui akurasi dari prediksi yang dihasilkan. Pengguna memasukkan data untuk diprediksi melalui *form* yang disediakan dan data tersebut ditelusuri dengan *decision tree*.

3.10. Perancangan Kelas

Perancangan kelas meliputi atribut dan operasi apa saja yang digunakan beserta algoritmanya. Berikut ini merupakan perancangan dari kelas – kelas yang digunakan:

Tabel 3 Perancangan kelas

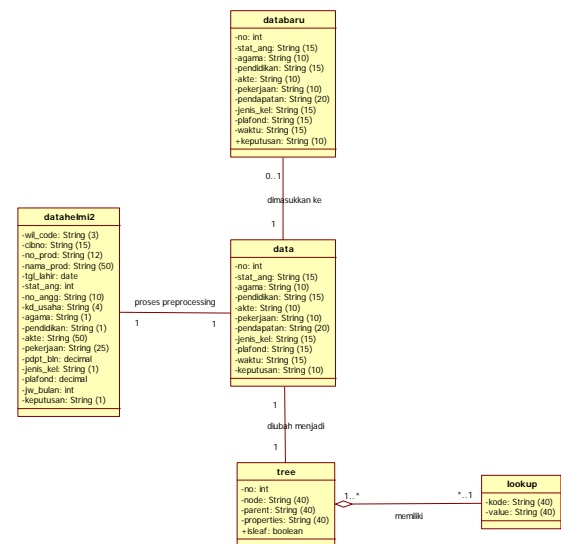
No	Nama Kelas	Atribut	Operasi
1	ID3		<i>Method</i> Konstruktor BuildTree SaveNewData
2	Node	Label Value Parent Properties IsLeaf	<i>Method</i> konstruktor <i>Method</i> selektor
3	Record	Sa Dat	<i>Method</i> konstruktor <i>Method</i> selektor
4	Data	Stat_ang Jenis_kel Agama Pendidikan Akte Pekerjaan Pendapatan Plafond Waktu Keputusan	<i>Method</i> konstruktor <i>Method</i> selektor
5	SetAtribut	Set Properties	<i>Method</i> konstruktor <i>Method</i> selektor
6	Atribut	Index NamaAtribut IsUsed	<i>Method</i> konstruktor <i>Method</i> selektor
7	ControllerID3		<i>Method</i> konstruktor Train SaveNewData
8	Prediktor		<i>Method</i> konstruktor Prediksi
9	ControllerPrediktor		<i>Method</i> konstruktor MakePredic

No	Nama Kelas	Atribut	Operasi
			tion
10	Preproses		Praproses <i>Method</i> konstruktor
11	ControllerPreproses		<i>Method</i> konstruktor DoPreproses
12	TreeVisualizer		<i>Method</i> konstruktor ViewTree FindChild

3.11. Relasi Kelas yang Menjadi Tabel

Kelas – kelas yang dijadikan tabel dalam basis data yaitu kelas Node yang akan menjadi tabel tree, kelas Data yang akan menjadi tabel data dan databaru. Tabel datahelmi2 merupakan tabel yang digunakan untuk menyimpan data awal. Tabel lain yang terbentuk yaitu tabel lookup yang digunakan untuk memberi nama pada atribut.

Kelas yang menjadi tabel dipetakan dan disajikan dalam diagram kelas yang mempunyai penegasan *multiplicity* dan nama relasi. Relasi kelas yang terbentuk yaitu kelas Datahelmi2 diproses melalui *preprocessing* menjadi kelas Data. Kelas Databaru dimasukkan ke dalam kelas Data untuk proses *training*. Kelas Data akan diolah menjadi kelas Tree. Sedangkan kelas Tree memiliki kelas Lookup yang berupa atribut – atribut yang ada di dalam *tree*. Gambar 8 merupakan diagram kelas yang melalui pemetaan menjadi tabel.



Gambar 8 Relasi kelas yang menjadi tabel

4. Implementasi dan Pengujian

4.1. Implementasi Kelas

Kelas – kelas diimplementasikan dalam dua bagian yaitu :

1. Rancangan kelas pada sub bab 3.10 diimplementasikan dengan menggunakan bahasa pemrograman Java.
2. Beberapa rancangan kelas juga diimplementasikan ke dalam basis data untuk keperluan masing – masing.

Adapun kelas – kelas yang diimplementasikan dalam basis data yaitu kelas Node yang akan menjadi tabel tree, kelas Data yang akan menjadi tabel data dan databaru. Tabel lain yang terbentuk yaitu tabel lookup yang digunakan untuk memberi nama pada atribut.

4.2. Implementasi Antarmuka

Implementasi antarmuka diimplementasikan dalam *form* yang dibentuk dari kelas JFrame dalam Java. Beberapa implementasi dari rancangan antarmuka ditunjukkan pada gambar 9, 10, dan 11.

Form ControllerForm merupakan *form* induk yang mengontrol seluruh proses dan perintah. Implementasinya ditunjukkan pada gambar 9.

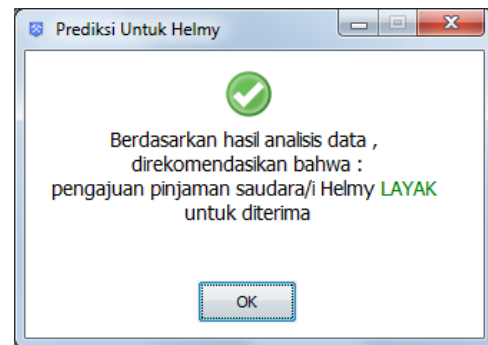


Gambar 9 Implementasi *form* ControllerForm

Form testing merupakan *form* yang digunakan untuk memprediksi data baru. Implementasinya ditunjukkan pada gambar 10.

Gambar 10 *Form* Testing

Form Popup digunakan untuk menampilkan notifikasi hasil prediksi. Implementasinya ditunjukkan pada gambar 11.



Gambar 11 *Form* Popup

4.3. Pengujian

Perancangan dan pelaksanaan pengujian dapat dilihat pada tabel 4.

Tabel 4 Tabel uji

No	Kebutuhan Fungsional	Butir Uji
1	Dapat melakukan pra-proses data	Menekan tombol data preprocessing atau menu item data preprocessing
2	Dapat membuat <i>decision tree</i> dengan algoritma ID3	Menekan tombol <i>data training</i> atau menu item <i>data training</i>
3	Dapat memberikan rekomendasi atau prediksi terhadap data baru	Menekan tombol PREDICT

No	Kebutuhan Fungsional	Butir Uji
		Memasukkan 20 <i>data set</i> yang telah disediakan dan menampilkan hasil prediksi
		Memasukkan 10 <i>data set</i> baru yang belum diketahui hasil prediksinya

Berdasarkan tiga kebutuhan fungsional pada sub bab 3.4 dilakukan pengujian dan menghasilkan beberapa hasil butir uji. Hasil pengujian dapat dilihat pada tabel 5.

Tabel 5 Hasil pengujian

No	Kebutuhan Fungsional	Hasil Yang Didapat
1	Dapat melakukan pra-proses data	Notifikasi pra-proses data muncul dan menu <i>data training</i> dan <i>data preprocessing</i> menjadi <i>enable</i>
2	Dapat membuat <i>decision tree</i> dengan algoritma ID3	Notifikasi muncul dan <i>decision tree</i> dapat ditampilkan dalam <i>form TreeViewer</i>
3	Dapat memberikan rekomendasi atau prediksi terhadap data baru	Hasil prediksi muncul dalam notifikasi
		Dari hasil <i>data testing</i> dengan 20 data, dihasilkan 17 data sesuai dengan harapan yang diprediksi
		Notifikasi hasil prediksi

4.4. Analisis Hasil Uji

Berdasarkan dari hasil uji pada tabel 4.8, dapat disimpulkan bahwa perangkat lunak dapat digunakan untuk :

1. Membuat *decision tree* dan menampilkan visualisasinya kepada pengguna. Visualisasi *decision tree* dituangkan dalam *form TreeViewer*.
2. Memprediksi data baru terkait dengan kelayakan permintaan pinjaman nasabah dengan akurasi 85 %. Prediksi dilakukan dengan

menerima masukan dari pengguna dan mengeluarkan hasil prediksi berdasarkan aturan dalam *decision tree*. Masukan diambil dari isian form oleh pengguna dan hasil prediksi dituangkan dalam notifikasi.

5. Kesimpulan dan Saran

5.1. Kesimpulan

Kesimpulan yang dapat diambil dari penelitian ini adalah :

1. Perangkat lunak ini dapat memberikan prediksi terhadap data permintaan pinjaman baru dengan akurasi sebesar 85 %. Prediksi dilakukan dengan mengklasifikasikan data baru ke dalam kelas layak atau tidak layak melalui *decision tree*.
2. Proses *retraining* dapat digunakan untuk mengatasi permasalahan perangkat lunak jika tidak dapat memberikan rekomendasi keputusan.

Hasil dari implementasi *data mining* ini dapat digunakan sebagai pertimbangan dalam mengambil keputusan lebih lanjut tentang kelayakan permintaan pinjaman dari nasabah.

5.2. Saran

Untuk pengembangan lebih lanjut, penulis memberikan beberapa saran yaitu :

1. Pada proses *data cleaning* disarankan menggunakan metode lain untuk mengatasi *noise* pada data kontinyu. Metode yang digunakan misalnya dengan penggantian dengan nilai yang sering muncul (modus) atau nilai rata – rata (*mean*).
2. Pada saat diskritisasi (pengubahan data kontinyu menjadi diskrit) disarankan menggunakan metode statistika, seperti *min – max normalization* atau *z-score standardization*.

REFERENSI

- [1] Campione Mary, dkk, 2000, "*Java Tutorial Third Edition : A Short Course On The Basic*", Addison Wesley, California.
- [2] Elmasri Ramez, dan B. Navathe Shamkan, 2000, "*Fundamental Of Database System, Third Edition*", Versaware Inc.
- [3] Han J., Kamber M., 2006, "*Data Mining : Concept and Technique, Second Edition*", Morgan Kaufmann, San Fransisco.
- [4] H. Witten Ian, Eibe Frank, 2005, "*Data Mining : Practical Machine Learning Tool and Techniques, Second Edition*", Morgan Kaufmann, San Fransisco.
- [5] Inmon, W. H., 1992, "*Building Data Warehouse*", Wiley, Indianapolis.
- [6] J. Cios K dkk, 2007, "*Data Mining, A Knowledge Discovery Approach*", Springer, New York.
- [7] Kusriani, Taufiq Luthfi Emha, 2009, "*Algoritma Data Mining*", ANDI offset, Yogyakarta.
- [8] Larose Daniel T., 2005, "*Discovering knowledge in Data : An Introduction to Data Mining*", John Willey & Sons Inc.
- [9] Lau Yun-Thung Ph.D, 2001, "*The Art Of Object, Object Oriented Design And Architecture*", Addison Wesley, California.
- [10] Loew, Doug, 2005, "*Java All In One Desk Reference For Dummies*", Wiley, Indianapolis.
- [11] Oestereich, Bernd, 2002, "*Developing Software With UML, Object Oriented Analysis And Design In Practice : Second Edition*", Pearson Education, London.
- [12] MacLennan J., Tang Z.H., 2005, "*Data Mining with SQL Server 2005*", Wiley, Indianapolis.
- [13] Pressman R. S., 1997, "*Software Engineering : A Practitioner's Approach*", McGraw-Hill Companies, Inc., New York.
- [14] Rainardi Vincent, 2010, "*Building a Data Warehouse With Example in SQL SERVER*", Apress Berkeley.
- [15] Santosa Budi, 2007, "*Data Mining : Teknik Pemanfaatan Data untuk Keperluan Bisnis*", Graha Ilmu, Yogyakarta.
- [16] Sholih, 2010, "*Analisis Dan Perancangan Berorientasi Obyek*", CV. Muara Indah, Bandung.